Upload Assignment: Assignment 8

ASSIGNMENT INFORMATION

Due Date	Points Possible
Wednesday, December 15, 2021 11:59 PM	**1.5**

Introduction

The purpose of this assignment is to write a report in the form of a journal or conference paper. It must contain an introduction, a theoretical part and experiments (whose guide is detailed below) and a section containing discussion and conclusion.

The work will be then structured as follows: an introduction will briefly talk about the Gaussian Process Networks for Machine Learning. It will cite the main references, what is important in Gaussian Processes and what is its novelty with respect to other approaches. It will cite the use and interpretation of the Kernel functions. Then, a theoretical section (outlined above) will detail the main aspects of the theory. Although you do not need to be creative in this section, use your own words and avoid paragraph copying in order to prove that you understand what you wrote. *Do not provide any code in this report.*

The review of references different from book [1] (that are in almost all cases cited in the book and available online) will help in this purpose.

Finally, a set of experiments and a conclusion will complete the work.

Theory

Linear Gaussian Processes

Briefly introduce the signal model and the optimization criterion used in Gaussian Processes, for the case of linear processes.

Gaussian processes in Recursive Kernel Hilbert Spaces

Provide the RKHS model for a Gaussian process.

Using the handbook, fully develop the dual expression for the posterior and the likelihood models of the Gaussian process.

Prove that the kernel matrix is actually a covariance matrix.

Inference over the hyperparameters

Provide a pseudocode of the optimization of the hyperparameters of a Gaussian Process using exact inference.

Experiments

Linear Gaussian Process

Experimental setup

Generate an ARMA process with moving average (MA) coefficients
$\mathbf{b} = \{0.0048, 0.0193, 0.0289, 0.0193, 0.0048\}^\top$
and auto-regressive coefficients
$\mathbf{a} = \{1, -2.3695, 2.3140, -1.0547, 0.1874\}^\top$.
Generate a process input $x[n]$ consisting of 100 samples ($1 \leq n \leq 100$) of Gaussian noise with unit variance and zero mean. Choose a uniformly distributed random subset of 20 samples of the process output $f[n]$ as training set.

This is the same structure as in lesson 7.3, slides 4 to 7.

Linear Gaussian Process for Prediction

Using Matlab or your favorite language, construct a linear Gaussian process for which the input is defined as $x[n] = n$, and the output is $f[n]$. Use always a dual (a.k.a. feature space) formulation in your code.

Represent the predicted values for $1 \leq n \leq 100$, together with its variance. Compare the result with the real values of the ARMA model obtained in the experimental setup.

Linear ARMA with AR(1) process noise

Construct an ARMA process equal to the previous one, and add now an AR(1) noise of parameter $a_n = 0.2$ process to the output $f[n]$, this is, to construct the output
$y[n] = f(n) + g[n]$
where $g[n] = w_g[n] + 0.2g[n-1]$ and $w_g[n]$ is white noise of variance 0.1.

Construct a model that exactly matches the process $f[n]$, this is
$f[n] = \mathbf{a}^\top \mathbf{f}[n-1] + \mathbf{b}^\top \boldsymbol{\omega}[n-1] = \mathbf{w}^\top \mathbf{x}[n]$
where $\mathbf{f}[n-1]$ and $\boldsymbol{\omega}[n-1]$ are vectors constructed with the $n-1$-th to $n-5$-th samples of the input and output processes.

Compute the theoretical value of the output covariance matrix. Use it to construct a Gaussian process to optimize parameters $\mathbf{w}$ using 100 samples of the data.

Compare your results with the actual parameters.

Test the model with another 100 samples of the data. Represent the predictive mean and the predictive mean ± 2 standar deviations.

Nonlinear Gaussian Process

If the ARMA model is approximated using a small FIR filter of the output, the expected performance will be poor. As an example, reproduce the previous experiment, but modifying the input so it only contains the last 5 values of $x[n]$, i.e., removing $\mathbf{f}[n]$ form the model.
Using a nonlinear model will improve the performance. Using the GPN software included in the book, construct a predictor of $y[n]$ that uses only the input values $\mathbf{x}[n]$ and a covariance matrix consisting of the sum of a square exponential plus a noise matrix. Let the software figure out the hyperparameters using the exact inference methods for two cases:

- With an output $f[n]$ corrupted by the white plus AR(1) noise of the section above.
- With only white noise

Explain the performance compared to the one of the previous section.

Discussion and conclusion

As in the previous exercises, both the theoretical developments as the experiments and their discussion will be evaluated, but the clarity, structure and organization of the paper will be assessed as well.
On the one hand, it is important to take into account that since the theory is given in the book, it will make significantly decrease the grading if poorlyexplained, but it will not significantly increase the grading if correctly exposed. On the other hand, the experimental part will modify the grade in both senses.

References

[1] Carl Edward Rasmussen and Chris Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006.